# SQL Server Interview Questions & Answers

This repository contains a comprehensive list of SQL Server interview questions and answers, including query examples, performance considerations, and common use cases. It is designed for job preparation, learning, and quick reference.

## Table of Contents

1. What is Normalization?

2. How to Implement Normalization?

3. What is Denormalization?

4. OLTP vs OLAP

5. 1st, 2nd, and 3rd Normal Forms

6. Primary Key vs Unique Key

7. What is a Foreign Key?

8. Primary Key vs Foreign Key

9. Char vs Varchar

10. Char vs NChar

11. Size of Char vs NChar

12. Use of Index

13. How Index Improves Search

14. Types of Indexes

15. Clustered vs Non-Clustered Index

16. Function vs Stored Procedure

17. What are Triggers?

18. Types of Triggers

19. After Trigger vs Instead Of Trigger

20. Use of Identity

21. Transactions and Implementation

22. What are Inner Joins?

23. Explain Left Join

24. Explain Right Join

25. Explain Full Outer Join

26. Explain Cross Join

27. Why Do We Need UNION?

28. Union vs Union All

29. Can Union Have Unequal Columns?

30. Can Union Have Different Data Types?

31. Aggregate Functions Used

32. When to Use Group By

33. Group By and Non-Grouped Columns

34. What is Having Clause?

35. Having vs Where Clause

36. Sorting Records

37. Default Sort Order

38. Removing Duplicates

39. Selecting Top N Records

40. Handling NULLs

41. Use of Wildcards

42. Use of Alias

43. Writing CASE Statements

44. Self Reference Tables

45. What is a Self Join?

46. BETWEEN Clause

47. What is a Subquery?

48. Subquery Returning Multiple Results

49. What is a Correlated Query?

50. Joins vs Subqueries

51. Performance: Joins vs Subqueries

52. Nth Highest Salary Using TOP

53. Nth Highest Salary Using Correlated Query

54. Nth Highest Salary Using OFFSET FETCH

55. Performance of All Nth Salary Methods

56. What is a CTE?

57. Can CTE be Reused?

58. Use of CTE

59. Recursive CTE Example

60. Real-World CTE Examples

61. Can CTE Perform Inserts/Updates?

62. Does CTE Update Physical Tables?

63. What are Temporary Tables?

64. Temp Tables vs CTE

65. Performance: CTE vs Temp Tables

66. What are Window Functions?

67. Partition Clause in Window Function

68. Group By vs Window Function

69. RANK vs DENSE\_RANK

70. Finding Duplicate Records

71. Finding Unique Records

72. Deleting Duplicates with ID

73. Deleting Duplicates Without ID

74. Deleting Duplicates Using CTE

Question 1 :- Explain normalization ?

Normalization is a database design technique to remove redundant data.

Question 2 :- How to implement normalization ?

Normalization is implemented by splitting tables in to two , one with reference data ( master table) and other transaction data.

Question 3 :- What is denormalization ?

Denormalization is a database design technique to improve search performance. In denormalization we merge table so that we need to fetch from less tables and thus increase search performance.

Question 4 :- Explain OLTP vs OLAP ?

OLTP Normalization avoids redundancy and we follow normalization design(1st,2nd and 3rd normal). OLAP Denormalization improve search performance and we follow denormalization design.

Question 5 :- Explain 1st,2nd and 3rd Normal form ?

1st normal form :- A table is in first normal form when the columns have Atomic values. It should not have repeating groups.

2nd Normal form :- First normal form should be satisfied. All non-key columns should be fully dependent on the Primary key.

3rd Normal :- All 1st and 2nd normal form should be satisfied. No transient dependency should be present.

Question 6 :- Primary Key vs Unique key ?

Remember the 2 Ns.

1st N NULLS :- Unique can have NULLS , but primary key can not have NULLS. 2nd N Numbers :- Many unique keys but only ONE Primary key.

Question 7 : What is Foreign key?

A foreign key is a column in one table that refers to the primary key in another table. It is used to establish a relationship between two tables. Unlike primary keys, foreign keys can contain duplicate and NULL values.

Question 8: Primary Key vs Foreign Key

• Primary key :It is used to uniquely find data .One table can have only one primary key. Primary key can not contain null values.

• Foreign key: It is used to build relationship between two tables.one table can have multiple foreign keys. Foreign key can contain duplicate and null values.

Question 9 :- Differentiate between Char vs Varchar ?

Char is Fixed length while varchar is variable length.

Question 10 :- Differentiate between Char vs NChar ?

If you want to just store English characters then use Char , For multilingual language (Non-English) use NChar.

Question 11 :- Whats the size of Char vs NChar ?

Char 1 Character = 1 byte , For NChar 1 Character = 2 bytes.

Question 12 :- What is the use of Index ?

Indexes increases search performance.

Question 13 :- How does it make search faster?

Search becomes faster because of Balance tree structure. Internally it creates Node and Leaf nodes to reach to the data quick.

Question 14:- What are the two types of Indexes ?

There are two types of indexes Clustered and Non-clustered Indexes.

Question 15 :- Clustered vs Non-Clustered index

In Clustered index leaf node will point to actual data. While in case of non-clustered index leaf node takes help of clustered index

Question 16 :- Function vs Stored Procedures

Function is used to return computed values but will not make any permanent changes to the environment. Only SELECT statements are allowed inside functions; INSERT, UPDATE, or DELETE operations are not permitted. Functions can be called from SELECT, WHERE, or even from other stored procedures. The output of a function is mostly a scalar value or a table-valued result.

A Stored Procedure is like a mini batch program. It can change the environment and supports INSERT, UPDATE, and DELETE operations. Stored procedures cannot be executed from a SELECT or WHERE clause or from inside other functions. They can return a single or multiple outputs.

In short:

A Function returns computed scalar values and cannot perform permanent changes (like insert, update, delete) inside it.

A Stored Procedure is a mini-program that can perform any operations like making changes to the database, running backups, and more.

Question 17 :- What are triggers and why do you need it ?

Triggers are logics which can be executed when events like insert,update,delete etc happens.

Question 18 :- What are types of triggers ?

There are two types After trigger and Instead OF trigger.

Question 19 :- Differentiate between After trigger vs Instead Of ?

After trigger :- After event has happened logic is executed. Instead Of trigger:- Instead of the event the logic is executed.

Question 20 :- What is need of Identity ?

Identity helps to define auto-incremented column.

Question 21 :- Explain transactions and how to implement it ?

Transaction treats series of activity as one single unit. Either everything is successful and or everything rollbacks.

Question 22 :- What are inner joins ?

Inner join selects matching records from both tables.

Question 23 :- Explain Left join ?

All data from left table selected and only matching records from right table.

Question 24 :- Explain Right join ?

All data from right table selected and only matching records from left table.

Question 25 :- Explain Full outer joins ?

All matching and unmatching records from both left and right table are selected.

Question 26 :- Explain Cross joins ?

Cross join is cartesian. Every record of one table is joined with other table records.

Question 27:-Why do we need UNION ?

Union combines two result sets and exclude duplicated.

select [ProductId], [ProductName]

from [LearnSql].[dbo].[mst\_Products]

union

select [ProductId], [ProductName]

from [LearnSql].[dbo].[mst\_ExpiredPoductsProducts]

Question 28:-Differentiate between Union vs Union All ?

Union combines result sets and excludes duplicates while Union all also combines result set but includes duplicates.

select [ProductId], [ProductName]

from [LearnSql].[dbo].[mst\_Products] union all

select [ProductId],

[ProductName]

from [LearnSql].[dbo].[mst\_ExpiredProducts]

Question 29:-can we have unequal columns in Union?

No. select

[ProductName]

from [LearnSql].[dbo].[mst\_Products] union all

select [ProductId],

[ProductName]

from [LearnSql].[dbo].[mst\_ExpiredProducts]

Question 30:-Can column have different data types in Union ?

No. select

[ProductName], [ProductId]

from [LearnSql].[dbo].[mst\_Products] union all

select [ProductId],

[ProductName]

from [LearnSql].[dbo].[mst\_ExpiredProducts]

Question 31:- Which Aggregate function have you used ?

Sum,Avg,Max, Min and Count.

select sum([CustomerAmount]), Avg([CustomerAmount]) , min([CustomerAmount]), max([CustomerAmount]), Count(\*)

from [dbo].[txn\_Customer]

Question 32:- When to use Group by ?

It helps to convert rows in to summary rows using common values. select [ProductName],sum([CustomerAmount])

from [dbo].[txn\_Customer] group by [ProductName]

Question 33:- Can we select column which is not part of group by ?

No. In a group by you can only select columns which are present in groupby.

Question 34:- What is having clause ?

Having clause helps to filter group by data.

select [ProductName],sum([CustomerAmount]) from [dbo].[txn\_Customer] group by [ProductName]

having [ProductName]='shoes'

Question 35:- Having clause vs Where clause

Sequence: In SQL, Where applies filters before the Group By clause is executed, while Having applies filters after the Group By clause.

Aggregate: Where cannot have aggregate functions, but Having can include aggregate functions like COUNT(), SUM(), AVG(), etc.

Filter Level: The Where clause filters data at the row level, whereas the Having clause filters data at the aggregate group level.

Question 36:- How can we sort records ?

Sorting is done by using order by clause. select \* from [dbo].[txn\_Customer] order by [CustomerAmount] desc

Question 37:- What’s the default sort ?

Ascending.

Question 38:- How can we remove duplicates ?

By using Distinct keyword.

SELECT DISTINCT

[ProductName]

FROM [dbo].[txn\_Customer];

Question 39:- Select the first top X records ?

By using the top keyword.

SELECT TOP 2 \*

FROM [dbo].[txn\_Customer];

Question 40:- How to handle NULLS ?

By using ISNULL function.

Question 41:- What is use of wild cards ?

Wild card helps in pattern matching.

Question 42:- What is the use of Alias ?

Alias helps to give different display names to original column names.

Question 43:- How to write a case statement ?

SELECT CustomerName,

CASE WHEN CustomerAmount < 200 THEN 'less than 200'

WHEN CustomerAmount > 200 THEN 'more than 200'

ELSE 'NA'

END AS CustomerCategory

FROM txn\_Customer;

Question 44:- What is self reference tables ?

Self reference tables are those tables who have primary key and foreign key in the same table.

Question 45:- What is self join ?

When you make joins ( inner,left,right) with same table it’s called as Self join.

SELECT

t1.Id,

t1.ReferenceId\_FK,

t2.CustomerName AS Name,

t1.CustomerName AS Reference

FROM

txn\_Customer AS t1

INNER JOIN

txn\_Customer AS t2

ON t1.ReferenceId\_FK = t2.Id;

Question 46:- Explain the between clause ?

Between clause helps to find values in between the range. select \* from txn\_Customer

where CustomerAmount between -200 and 200

Question 47:-Explain SubQuery ?

Subquery is a query inside a query (nested query).In Subquery first the inner query gets evaluated and then outer query.

Question 48:-Can inner Subquery return multiple results ?

Yes , Inner query can return multiple results but then in where clause you will need to us the “IN”

keyword.

Question 49:-What is Co-related Query ?

In co-related query first the outer query sends records to the inner query , inner query then evaluates and sends its back to the outer query.

Question 50:-Differentiate between Joins and SubQuery ?

Subquery Join

Intention Series of processing where one processing sends output to

other. Join two tables and get matching or not matching

records.

Select fields Can not select from inner

Query Can select multiple fields from

table.

Question 51: -Performance Joins vs Subquery?

Most of the times Joins should perform better. But not necessarily, its possible subquery can be faster many times. So SQL plan needs to be looked in to determine whose performance can be better.

Question 52:- Select the top nth highest salary using top and order by?

SELECT TOP 1 \* FROM (

SELECT TOP 2 EmpSalary FROM tblEmployee ORDER BY EmpSalary DESC

) AS innerquery ORDER BY EmpSalary;

Question 53:- Select the top nth highest salary using correlated Queries?

SELECT o1.EmpSalary

FROM tblEmployee AS o1

WHERE 3 = (SELECT COUNT(\*) FROM tblEmployee AS i1

WHERE o1.EmpSalary <= i1.EmpSalary);

Question 54:- Select top nth using using TSQL

SELECT DISTINCT EmpSalary FROM Employee

ORDER BY EmpSalary DESC

OFFSET 2 ROWS FETCH NEXT 1 ROW ONLY;

Question 55:- Performance comparison of all the methods.

TOP + ORDER BY is good for small N. Correlated subquery is platform-portable but slow. OFFSET FETCH is modern but needs SQL Server 2012+.

Question 56 :- What is CTE ?

CTE stands for common table expression. CTE is a TEMPRORAY RESULT SET which can be used in the

SUBSEQUENT execution scope ONLY ONCE.

Question 57 :- Can we execute CTE multiple times ?

After creation of CTE ,in the same subsequent context you can execute multiple times. If the context changes you can not use it any more.

Question 58 :- What is the use of CTE ?

CTE is used for writing a recursive query.

Question 59 :- How to write a recursive CTE ?

with countup as (

select 1 as n union all

select n+1 from countup where n<3

)

select \* from countup

Question 60 :- Can we see some real world examples of CTE ?

1) Employee Hierarchy

2) Nth Highest Salary

3) Filter & Reuse

4) Cleanup

5) Simplify logic

6) Recursive operations

Question 61 :- Can we perform insert updates on CTEs ?

Yes, but CTE must be updatable and not recursive for that.

Question 62 :- Does is update the tables physically ?

Yes, CTE updates affect the underlying physical tables

Question 63 :- What are temporary tables ?

Temporary tables are physical tables which gets created for a session. Once the session is closed temporary table is dropped.

CREATE TABLE #tblproducts ( product\_name VARCHAR(MAX), list\_price DEC(10,2)

);

insert into #tblproducts values('Shoes',100.23) select \* from #tblproducts

Question 64:- Temp tables vs CTE

Where are they created? Temporary tables are created physically inside SQL Server, whereas CTEs (Common Table Expressions) are created in memory.

What’s the lifetime? Temporary tables exist until the session is closed. In contrast, a CTE only exists within the subsequent context in which it is defined.

When to use? Temporary tables are best used when you need to store intermediate or temporary data for the session. CTEs are ideal for writing recursive queries or simplifying complex queries in a single-use context.

Question 65: - Performance CTE vs Temp

Temporary tables are physical tables which gives us the full power of normal tables, like we can create clustered / non-clustered indexes, primary keys and so on. While CTE are in memory temporary result set. So CTE does not have any way to access these performance tools.

So from that thought process Temp tables should perform better than CTE in most cases.

Question 66 :- What are window functions in SQL?

Let’s first understand these two words “Window” and “Function” in SQL Server context.

Window: - It represents set of rows.

Function: - Logic / Calculation which operates over that set of rows.

“Window function” is nothing but function (logic) which operates on group/set of rows and does calculations.

Syntax of windows function for “row\_number” looks as below. Read this line and see the bold part in the code below:- “Apply the row\_number over the column name order by ascending”. Ascending

is the default sort.

SELECT

ROW\_NUMBER() OVER (ORDER BY name) AS Id,

name

FROM tblNames;

Question 67 :- What does partition clause in window function ?

Partition clause in windows function divides rows and function in separate windows (groups) and applies function calculations on those groups.

Question 68 :- So is window function partition similar to group by ?

No, they are very much different.

Group By vs Window Function

The Group By clause combines multiple rows into a single row per group based on the specified column(s). It performs aggregation operations such as SUM, COUNT, or AVG on each group. As a result of this grouping, the original individual rows are lost.

In contrast, a Window Function performs calculations across a set of rows related to the current row without collapsing them. It allows running calculations like rankings, running totals, or moving averages, while keeping all original rows intact. This means Window Functions are ideal for performing aggregated calculations without losing the row-level detail.

Question 69 :- What is difference between Rank vs Dense\_Rank ?

Both these functions are only applicable when you have TIES.

RANK() leaves gaps in ranking when there are ties, while DENSE\_RANK() gives consecutive ranks without gaps

Question 70 :- Find Duplicate Records from a table.

SELECT name

FROM [CustomerDB].[dbo].[tblNames]

GROUP BY name

HAVING COUNT(name) > 1;

Question 71 :- Find Records which are unique.

SELECT name

FROM [CustomerDB].[dbo].[tblNames]

GROUP BY name

HAVING COUNT(name) = 1;

Question 72:- Delete Duplicate records (With Id)

When you get these kinds of questions it’s important to define the approach. If you have a proper approach in your mind then you can answer if there are tweaks in the questions. For this question three step approach should be used

Step 1: - Make sure that the base tables

has unique ID’s representing the rows.

Step 2: - Get Unique records with their ID’s

( Using Group by mostly)

Step 3:- Delete records other than records which have come from Step 2.

So the Final SQL query comes up something like this.

DELETE FROM tblNames

WHERE Id NOT IN (

SELECT MIN(Id)

FROM [CustomerDB].[dbo].[tblNames]

GROUP BY name

);

Question 73 :- Delete Duplicate records ( With out Id)

DELETE basetable

FROM (

SELECT

ROW\_NUMBER() OVER (PARTITION BY name ORDER BY name) AS rn,

name

FROM tblNames

) AS basetable

WHERE rn > 1;

Question 74 :- Delete Duplicate records ( using CTE)

WITH RankedNames AS ( SELECT

ROW\_NUMBER() OVER (PARTITION BY Name ORDER BY Name) AS Idnew,

Name

FROM tblNames

)

DELETE FROM RankedNames WHERE Idnew>1;